# ECSE 551 Mini-Project 1: Logistic Regression with Gradient Descent Algorithm

**Shankhin Brahmavar**
260921778
shankhin.brahmavar@mail.mcgill.ca

## Abstract

In this project we investigated the performance of linear classification models on two benchmark datasets. We used 2 different algorithms - a regular Logistic Regression algorithm and another Logistic Regression algorithm with an additional bias term - applied on 3 different datasets - one which was originally given to us, one that had the feature with the lowest absolute coefficient value deleted and another that had the feature with the highest absolute coefficient value squared and added as an extra feature to the dataset. This inturn gave us 6 models to train and validate using a 10-fold cross-validation algorithm. We found differences in the accuracy between each of these models, although the differences in these accuracy numbers varied in the ranges of 0-5% depending on the training and validation sets chosen, these are still an indicator of higher performance. In turn we found that the accuracy in general followed the trend of: LR with bias > regular LR and LR($x_i^2$) > LR with original dataset > LR($x_{ideleted}$).

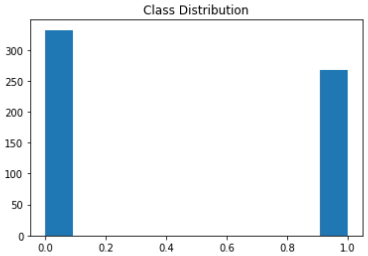
## 1 Introduction

We were tasked with finding the best linear regression model for the given datasets using the gradient descent algorithm. We were given to datasets - one of which was a dataset of 600 patients with 8 medical predictor variables and were tasked with classifying whether the said patient had diabetes(1) or not (0) based on these variables and the other was a dataset of 1599 red wine samples with 11 physicochemical features and were tasked with classifying whether the given sample was a high quality sample (1) or not(0). We observed that a Logistic Regression algorithm with bias that used a dataset that had the feature with the highest absolute coefficient value squared and added as an extra feature to the dataset performed the best and a regular Logistic Regression algorithm with the removal of the feature with the lowest absolute coefficient value performed the worst, although only slightly less than a regular Logistic Regression algorithm run on the original dataset.

## 2 Datasets

The diabetes dataset was a 600X8 matrix with 600 examples and 8 features whereas the red wine dataset was a 1599X11 matrix with 1599 examples and 11 features. We observed that in the diabetes dataset, there were a higher occurrence on 0s i.e a higher number of patients without diabetes as observed in the histogram in figure 1. Similarly, we observed that in the red wines dataset, there were a higher occurence of 1's i.e a higher number of samples that were a high quality wine sample as observed in figure 2. We came up with just one new feature for each dataset which was just the the feature with the highest absolute coefficient value squared since these features showed high correlation to whether a patient had diabetes or not or if

Figure 1: Class distribution of diabetes dataset



34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

Figure 2: Class distribution of red wine dataset

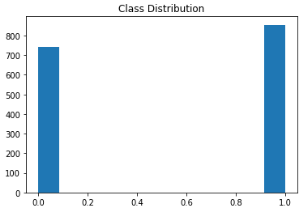
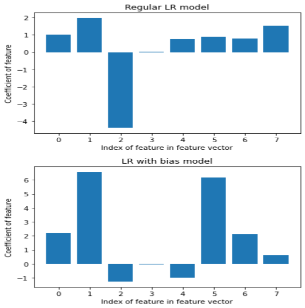


Figure 3: Index of diabetes feature vs Weights



a wine sample was high quality or not. We also ran a dataset with deletion of the feature with the lowest absolute coefficient value as this feature would have little to no correlation to whether a patient had diabetes or not or if a wine sample was high quality or not. The coefficient values of each feature for both datasets are represented with a bar graph in figure 3 and figure 4.

## 3 Results

When running the algorithm, we observed that the LR with bias algorithm in general ran better better than the regular LR algorithm as can be seen by the cross-entropy loss vs number of iterations curve for the diabetes dataset in figure 5. Through this curve it was also observed that as we increase number of iterations, there is little to no change in the cross-entropy loss with a significant increase in execution time. Hence, we kept the number of iteration at 10000 for both. We also observed through the learning rate vs accuracy curve in figure 6 that a certain range of learning rate (here generally around 1.5-3) favoured a higher accuracy for a set number of iterations for the red wine dataset. We also observed that addition of a feature increased test accuracy - a sample run on the diabetes dataset is given in table 1.

Figure 4: Index of red wine feature vs Weights

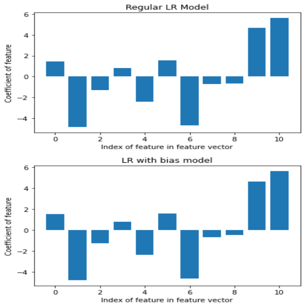
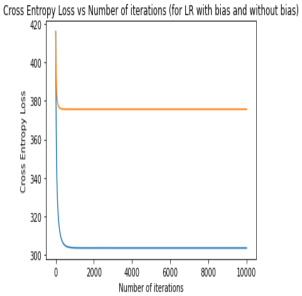


Table 1: Accuracy for each model on a sample run

| | Part | |
|---|---|---|
| Dataset | Accuracy(LR,LR with bias) | Ranking |
| Diabetes added feature | 69.166,69.333 | 1 |
| Diabetes original | 66.166,64.591 | 2 |
| Diabetes deleted | 62.000,62.666 | 3 |

Figure 5: number of iterations vs CEL



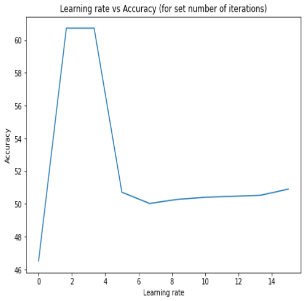
## 4 Discussion and Conclusion

We concluded that addition of a feature that is related to the feature with the highest weights increases the accuracy of the Logistic Regression algorithm as well as adding a bias term to the Logistic Regression algorithm results in increasing accuracy of the model. If the features weren't scaled to be between 0 and 1 we would have needed to perform normalization of the features as if this had not been done would result in higher value features dominating lower values ones. Additionally, either L1 regularization or L2 regularization of the logistic regression model are possible directions for further investigation into increasing the accuracy of a logistic regression algorithm. Optimization of the algorithm in terms of F1-score can also be looked into.

## 5 Statement of Contributions

Since I was the only one in my group, I did all of the work and research myself.

Figure 6: Learning Rate vs Accuracy



# 6 Appendix

## 6.1 FinalDiabetes.ipynb

The code can be found at the following link: https://colab.research.google.com/drive/16rQbzG-JVjUFe1Gc_UMw2nrDLXht_K7r?usp=sharing

## 6.2 FinalRedWine.ipynb

https://colab.research.google.com/drive/1o8YDa0BMh0QGKouceiciUyoc5lkJ9tuy?usp=sharing